**DAMG 7275 Project Team 8**

**Team Member**

1. Jiayue Han - NUID: 002965004

2. Zhiqing Wang - NUID: 002916484

3. Kewhi Shi - NUID: 001359675

4. Xinan Wang - NUID:002916472

**Topic**

US Used Car Data Management System

**Implementation Process**

1. Data pre-processing using Python

Dataset for this project is from Kaggle.com, so we did some preliminary cleansing by dropping several columns that do not contain useful information and sorting the dataset by sales id. For testing purpose and because of the long runtime due to size of the original file, we used a fragment of the dataset to test our data transformation and ingestion method for now. Below is a screenshot of our test data.

Table

Description automatically generated

1. Data storage using Azure Blob Storage

Two containers were created to store input file in .CSV format and output file in. JSON format. Below are screenshots of our test data before and after transformation.

Before:

Graphical user interface, text, application

Description automatically generated

After:

Graphical user interface, text, application

Description automatically generated

1. Data transformation using Data Flows in Azure Data FactoryGraphical user interface

   Description automatically generated with medium confidence

To transform data from .CSV format into. JSON format, and ingesting data into CosmosDB, we first used the csv file stored in blob storage as the source file. The next step was to create sub documents using custom columns. We implemented this concept by creating a sub column called ‘sale’ and included sales region and sales price. We will create more sub documents to categorize data into sales information, used car information, etc., The final step in data flow was to output file in. JSON format and store it in the output container in the blob storage.

1. Data ingestion into CosmosDB

Graphical user interface, text, chat or text message

Description automatically generated with medium confidence

For the data pipeline, since we are manually uploading original data to blob storage, the first step of the pipeline is transforming data using data flows mentioned in the previous step. On completion, the json file would be ingested into CosmosDB. The source was the transformed file, and the sink dataset was our project CosmosDB SQL API database. Below is the screenshot of ingested data. From this point, we will explore scheduled update of the input data and integration with graph database using CosmosDB for Gremlin.

Graphical user interface, application

Description automatically generated